

**IN THE CLAIMS**

Please amend the claims as follows:

Claim 1 (Currently amended): A concurrency control method in a transaction processing system for processing a plurality of transactions in parallel with respect to hierarchical data, the concurrency control method comprising:

producing a copy of the hierarchical data at a time of starting an access to the hierarchical data by each transaction;

judging whether a collision between one of reading access or writing access to be made by a first transaction with respect to a copy of the hierarchical data for the first transaction and another one of the reading access or writing access made by the second transaction with respect to a copy of the hierarchical data for the second transaction will occur or not, when the first transaction and the second transaction are accepted at the same time as concurrent transactions for accessing the same location of the hierarchical data;

carrying out a processing for avoiding the collision due to the concurrent transactions when the judging step judges that the collision will occur;

and reflecting a writing access made by the first transaction with respect to a copy of the hierarchical data for the first transaction, on the hierarchical data, when the first transaction is to be finished normally, and reflecting the writing access also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet.

Claim 2 (Original): The concurrency control method of Claim 1, wherein the judging step judges whether the collision will occur or not, according to whether data looked up by making the reading access without taking the writing access into consideration and data looked up by making the reading access by taking the writing access into consideration are identical or not.

Claim 3 (Original): The concurrency control method of Claim 1, wherein when the first transaction is to make the reading access with respect to a copy of the hierarchical data, the judging step judges whether the collision will occur or not according to whether first data looked up by making the reading access with respect to a copy of the hierarchical data for the first transaction and second data looked up by making the reading access with respect to data obtained by merging a copy of the hierarchical data for the first transaction and a copy of the hierarchical data for the second transaction are identical or not.

Claim 4 (Original): The concurrency control method of Claim 3, wherein the judging step judges that the collision will not occur when the first data and the second data are judged as identical for all transactions that can be the second transaction, and judges that the collision will occur otherwise.

Claim 5 (Original): The concurrency control method of Claim 1, further comprising:  
making the writing access with respect to a shared copy produced by copying the hierarchical data in order to reflect writing accesses made by all transactions that make accesses to the hierarchical data, when the first transaction is to make the writing access with respect to a copy of the hierarchical data;

wherein when the first transaction is to make the reading access with respect to a copy of the hierarchical data, the judging step judges whether the collision will occur or not according to whether first data looked up by making the reading access and second data looked up by making the reading access with respect to the shared copy of the hierarchical data are identical or not.

Claim 6 (Original): The concurrency control method of Claim 5, wherein the judging step judges that the collision will not occur when the first data and the second data are judged as identical, and judges that the collision will occur when the first data and the second data are judged as not identical.

Claim 7 (Original): The concurrency control method of Claim 5, wherein when there is an upper limit to a number of shared copies that can be recorded, those shared copies which have a higher possibility of being utilized at a time of reproducing a state in which the reading access is to be made later on are recorded at a higher priority, among the shared copies corresponding to states at times of the writing accesses with respect to the hierarchical data.

Claim 8 (Original): The concurrency control method of Claim 1, wherein when the first transaction is to make the writing access with respect to a copy of the hierarchical data, the judging step judges whether the collision will occur or not according to whether first data looked up by making the reading access of the second transaction and second data looked up by making the reading access of the second transaction with respect to a state of the hierarchical data after the writing access are identical or not, for all reading accesses by all transactions that make accesses to the hierarchical data and that can be the second transaction.

Claim 9 (Original): The concurrency control method of Claim 8, wherein the judging step judges that the collision will not occur when the first data and the second data are judged as identical for all reading accesses of all transactions that make accesses to the hierarchical data and that can be the second transaction, and judges that the collision will occur otherwise.

Claim 10 (Original): The concurrency control method of Claim 8, further comprising:  
recording an access sequence of accesses made with respect to a copy of the  
hierarchical data by each transaction, for each one of all transactions that make accesses to  
the hierarchical data;

wherein the judging step obtains all reading accesses of all transactions that make  
accesses to the hierarchical data and that can be the second transaction, by looking up a  
record of the access sequence.

Claim 11 (Original): The concurrency control method of Claim 8, further comprising:  
recording data looked up by making the reading accesses;  
wherein the judging step obtains the first data by looking up a record of the data  
looked up.

Claim 12 (Original): The concurrency control method of Claim 8, wherein the  
judging step obtains the first data as data obtained by making the writing access that was  
made by the second transaction before the reading access with respect to a state of the  
hierarchical data at a start of the second transaction, and then making the reading access with  
respect to a state of the hierarchical data after the writing access.

Claim 13 (Original): The concurrency control method of Claim 8, further comprising:  
making the writing access with respect to a shared copy produced by copying the  
hierarchical data in order to reflect writing accesses made by all transactions that make  
accesses to the hierarchical data, when the first transaction is to make the writing access with  
respect to a copy of the hierarchical data;

and storing states of the shared copy at timings at which the writing accesses were made by some of the transactions that make accesses to the hierarchical data;

wherein the judging step obtains the first data as data obtained by reproducing a state of the hierarchical data at a timing at which the reading access was made by selecting one of stored states of the shared copy which is close to the state of the hierarchical data at a timing at which the reading access was made and making the writing access that was made by the second transaction with respect to a selected state of the shared copy according to need, and then making the reading access with respect to a reproduced state of the hierarchical data.

Claim 14 (Original): The concurrency control method of Claim 13, wherein when there is an upper limit to a number of shared copies that can be recorded, those shared copies which has a higher possibility of being utilized at a time of reproducing a state in which the reading access is to be made later on are recorded at a higher priority, among the shared copies corresponding to states at times of the writing accesses with respect to the hierarchical data.

Claim 15 (Original): The concurrency control method of Claim 8, wherein the judging step obtains the second data as data obtained by making the writing access of the second transaction with respect to a state after the writing access was made with respect to a copy of the hierarchical data for the first transaction, and then making the reading access with respect to a state of the hierarchical data after the writing access of the second transaction.

Claim 16 (Original): The concurrency control method of Claim 8, further comprising:  
making the writing access the respect to a shared copy produced by copying the hierarchical data in order to reflect writing accesses made by all transactions that make

accesses to the hierarchical data, when the first transaction is to make the writing access with respect to a copy of the hierarchical data;

and storing states of the shared copy of timings at which the writing accesses were made by some of the transactions that make accesses to the hierarchical data;

wherein the judging step obtains the second data as data obtained by reproducing a state of the hierarchical data at a timing at which the reading access is to be made by selecting one of the stored states of the shared copy which is close to the state of the hierarchical data at a timing at which the reading access is to be made, making the writing access that was made by the first transaction after that timing, with respect to a selected state of the shared copy, and making the writing access that was made by the second transaction according to need, and then making the reading access with respect to a reproduced state of the hierarchical data.

Claim 17 (Original): The concurrency control method of Claim 16, wherein when there is an upper limit to a number of shared copies that can be recorded, those shared copies which have a higher possibility of being utilized at a time of reproducing a state in which the reading access is to be made later on are recorded at a higher priority, among the shared copies corresponding to states at times of the writing accesses with respect to the hierarchical data.

Claim 18 (Original): The concurrency control method of Claim 1, wherein when the judging step judges that the collision will occur, the carrying out step carries out the processing for keeping those transactions that are determined according to prescribed criteria among transactions related to the collision, to wait until other transactions related to the collision are finished.

Claim 19 (Currently amended): A transaction processing system for processing a plurality of transactions in parallel with respect to hierarchical data, comprising:

a copying unit configured to produce a copy of the hierarchical data at a time of starting an access to the hierarchical data by each transaction;

a judging unit configured to judge whether a collision between one of reading access or writing access to be made by a first transaction with respect to a copy of the hierarchical data for the first transaction and another one of reading access or writing access made by the second transaction with respect to a copy of the hierarchical data for the second transaction will occur or not, when the first transaction and the second transaction are accepted at the same time as concurrent transactions for accessing the same location of the hierarchical data;

a processing unit configured to carry out a processing for avoiding the collision due to the concurrent transactions when the judging unit judges that the collision will occur;

and a reflecting unit configured to reflect a writing access made by the first transaction with respect to a copy of the hierarchical data for the first transaction, on the hierarchical data, when the first transaction is to be finished normally, and reflect the writing access also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet.

Claim 20 (Currently amended): A computer program product which employs a storage medium for causing a computer to function as a transaction processing system for processing a plurality of transactions in parallel with respect to hierarchical data, the computer program product comprising:

a first computer program code loaded in a processor for causing the computer to produce a copy of the hierarchical data at a time of starting an access to the hierarchical data by each transaction;

a second computer program code loaded in a processor for causing the computer to judge whether a collision between one of reading access or writing access to be made by a first transaction with respect to a copy of the hierarchical data for the first transaction and another one of reading access or writing access made by the second transaction with respect to a copy of the hierarchical data for the second transaction will occur or not, when the first transaction and the second transaction are accepted at the same time as concurrent transactions for accessing the same location of the hierarchical data;

a third computer program code loaded in a processor for causing the computer to carry out a processing for avoiding the collision due to the concurrent transactions when the second computer program code judges that the collision will occur;

and a fourth computer program code loaded in a processor for causing the computer to reflect a writing access made by the first transaction with respect to a copy of the hierarchical data for the first transaction, on the hierarchical data, when the first transaction is to be finished normally, and reflect the writing access also on a copy of the hierarchical data for the second transaction if the second transaction is not finished yet.

Claim 21 (New): A computer program product which employs a storage medium for causing a computer to function as a transaction processing system for processing a plurality of transactions in parallel with respect to hierarchical data, the computer program product comprising:

a first computer program code loaded in a processor for causing the computer to accept transactions which are temporarily overlapping;

a second computer program code loaded in a processor for causing the computer to produce a copy of the hierarchical data at a time of starting execution of each transaction;



a third computer program code loaded in a processor for causing the computer to judge whether or not the copy of a first transaction will conflict with the copy of a second transaction after execution of the first and second transaction;

a fourth computer program code loaded in a processor for causing the computer, when it is judged that a conflict will occurs between the first and second transactions, to halt the execution of the second transaction until the execution of the first transaction has been completed; and

a fifth computer program code loaded in a processor for causing the computer to reflect the copy of the first transaction in the hierarchical data, when the first transaction is finished normally, and reflect the copy of the first transaction in the copy of the second transaction, when the second transaction is not finished yet.